

Windows Programming

Visual C++ MFC Programming

Lecture 06

김예진

Dept. of Game Software

Notices

- 03/07: 502 → 501 등록 이동
- 03/21: HW 1 (Due: 03/28)
- 04/09: HW 2 (Due: 04/16)

Plan

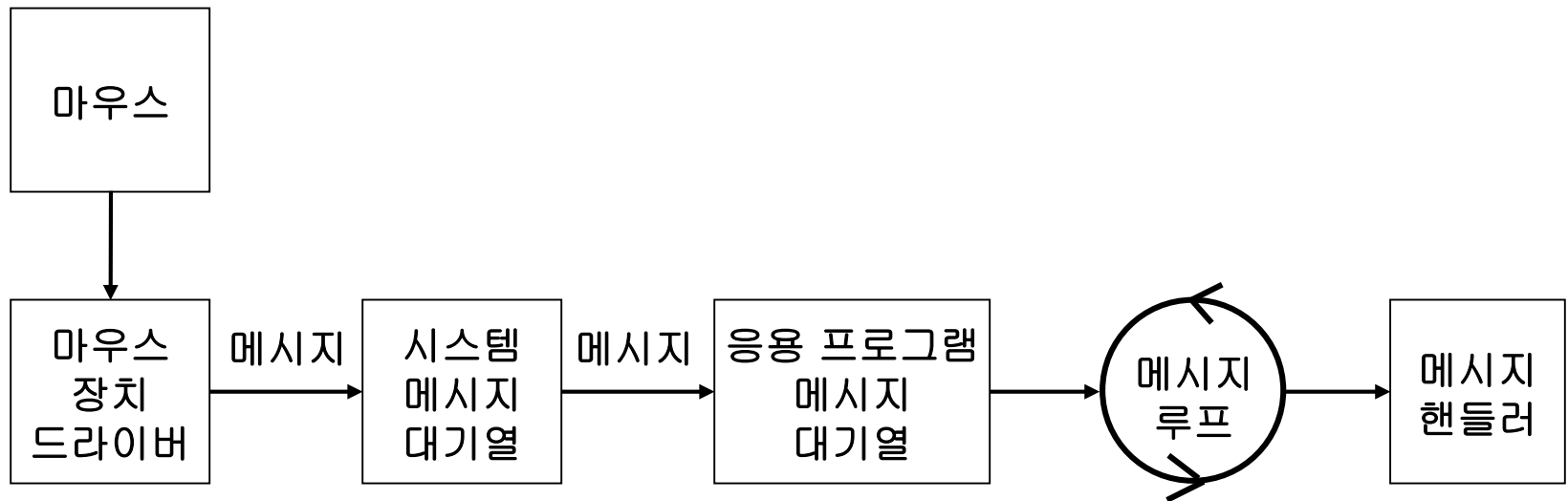
- Mouse input
- Keyboard input

MFC와 Mouse



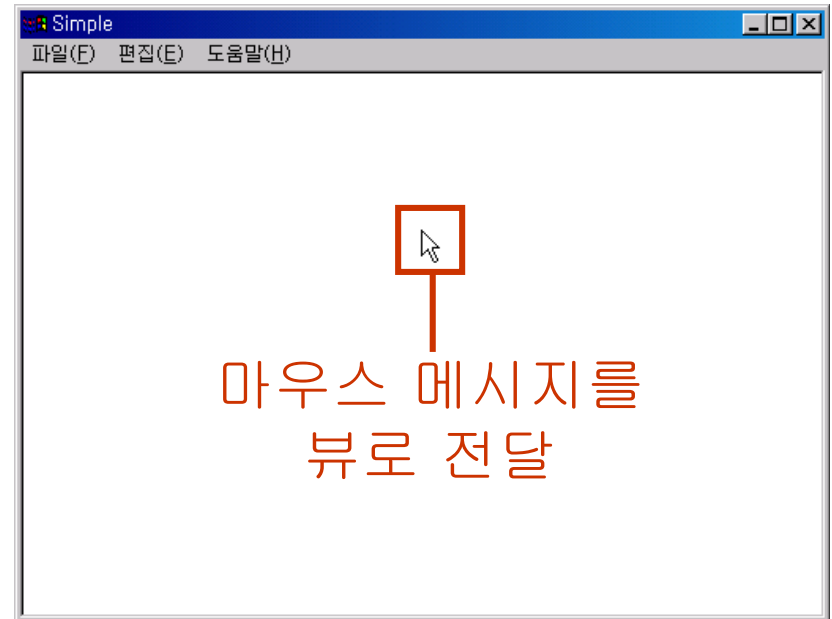
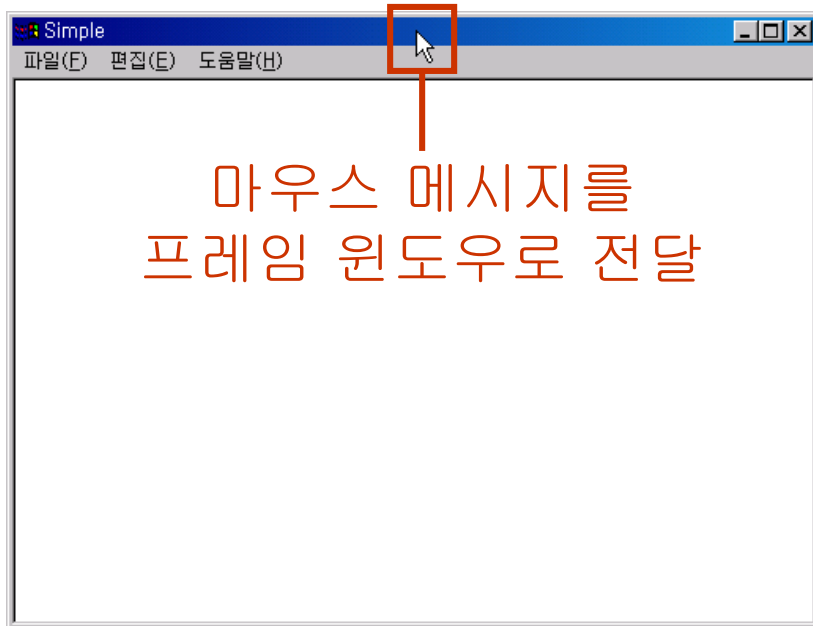
Mouse Input

- Mouse 처리
 - 윈도우 운영체제는 mouse와 관련된 모든 변화를 message 형태로 program에게 전달



Mouse Input

- Mouse message 전달
 - Mouse message는 mouse cursor 밑에 있는 윈도우가 받음



Mouse Input

- Client 영역 mouse message

메시지	발생 시점
WM_LBUTTONDOWN	왼쪽 버튼을 누를 때
WM_LBUTTONUP	왼쪽 버튼을 떼를 때
WM_LBUTTONDOWNBLCLK	왼쪽 버튼을 더블 클릭할 때
WM_MBUTTONDOWN	가운데 버튼을 누를 때
WM_MBUTTONUP	가운데 버튼을 떼를 때
WM_MBUTTONDOWNBLCLK	가운데 버튼을 더블 클릭할 때
WM_RBUTTONDOWN	오른쪽 버튼을 누를 때
WM_RBUTTONUP	오른쪽 버튼을 떼를 때
WM_RBUTTONDOWNBLCLK	오른쪽 버튼을 더블 클릭할 때
WM_MOUSEMOVE	마우스를 움직일 때

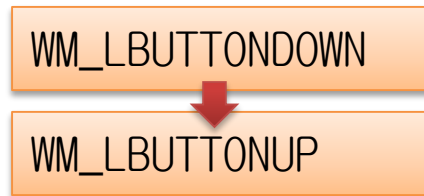
Mouse Input

- Message 발생 과정 (왼쪽 마우스 버튼의 경우)

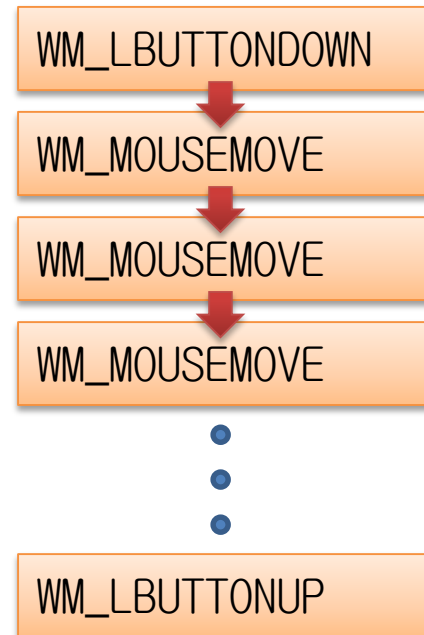
마우스를 움직일 때



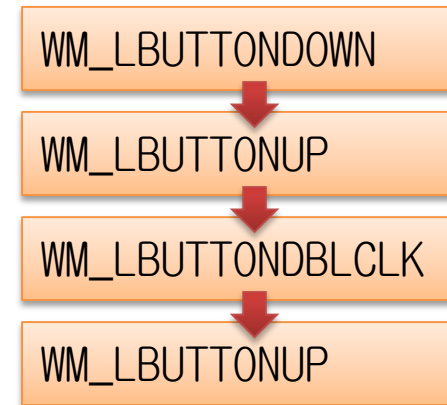
왼쪽 버튼 클릭



왼쪽 버튼 클릭
+ 드래그



왼쪽 버튼
더블 클릭



Mouse Input

- Client 영역 mouse message handler

메시지	메시지 맵 매크로	메시지 핸들러
WM_LBUTTONDOWN	ON_WM_LBUTTONDOWN()	OnLButtonDown
WM_LBUTTONUP	ON_WM_LBUTTONUP()	OnLButtonUp
WM_LBUTTONDBLCLK	ON_WM_LBUTTONDBLCLK()	OnLButtonDblClk
WM_MBUTTONDOWN	ON_WM_MBUTTONDOWN()	OnMButtonDown
WM_MBUTTONUP	ON_WM_MBUTTONUP()	OnMButtonUp
WM_MBUTTONDBLCLK	ON_WM_MBUTTONDBLCLK()	OnMButtonDblClk
WM_RBUTTONDOWN	ON_WM_RBUTTONDOWN()	OnRButtonDown
WM_RBUTTONUP	ON_WM_RBUTTONUP()	OnRButtonUp
WM_RBUTTONDBLCLK	ON_WM_RBUTTONDBLCLK()	OnRButtonDblClk
WM_MOUSEMOVE	ON_WM_MOUSEMOVE()	OnMouseMove

Mouse Input

- Message handler 형태 (1/3)

```
afx_msg void On##### (UINT nFlags, CPoint point);
```

- nFlags: Message가 생성될 때 mouse나 keyboard button의 상태를 나타내는 bit mask

비트 마스크	의미
MK_CONTROL	Ctrl 키가 눌렸을 때
MK_SHIFT	Shift 키가 눌렸을 때
MK_LBUTTON	마우스 왼쪽 버튼이 눌렸을 때
MK_MBUTTON	마우스 가운데 버튼이 눌렸을 때
MK_RBUTTON	마우스 오른쪽 버튼이 눌렸을 때

Mouse Input

- Message handler 형태 (2/3)

```
afx_msg void On##### (UINT nFlags, CPoint point);
```

– nFlags 사용 예

```
void CChildView::OnLButtonDown(UINT nFlags, CPoint point)
{
    if (nFlags & MK_SHIFT) {
        // 만약 Shift 키가 눌렸다면 ...
    }

    CWnd::OnLButtonDown(nFlags, point);
}
```

Mouse Input

- Message handler 형태 (3/3)

```
afx_msg void On##### (UINT nFlags, CPoint point);
```

- point: Message가 생성될 때 mouse cursor(pointer) 위치
 - Client area 윈도우 좌표계 사용
- point 사용 예



```
void CChildView::OnLButtonDown(UINT nFlags, CPoint point)
{
    CClientDC dc(this);

    CPoint pt = point;
    dc.Rectangle(pt.x - 100, pt.y + 100, pt.x + 100, pt.y - 100);

    CWnd::OnLButtonDown(nFlags, point);
}
```

Mouse Input

- Mouse capture

- 문제점: Mouse가 client 윈도우 밖으로 나가 버리면?
- 용도: Mouse capture를 하면 mouse cursor의 위치에 관계없이 mouse message를 받을 수 있음
- 관련 함수

API 함수	MFC 함수	의미
SetCapture()	CWnd::SetCapture()	마우스 캡처를 시작한다.
ReleaseCapture()	없음	마우스 캡처를 해제한다.
GetCapture()	CWnd::GetCapture()	어느 윈도우가 현재 마우스 캡처를 하고 있는지 알아낸다.

Mouse Input

- 비 client 영역 mouse message

메시지	발생 시점
WM_NCLBUTTONDOWN	왼쪽 버튼을 누를 때
WM_NCLBUTTONUP	왼쪽 버튼을 떼를 때
WM_NCLBUTTONDOWNDBLCLK	왼쪽 버튼을 더블 클릭할 때
WM_NCMBBUTTONDOWN	가운데 버튼을 누를 때
WM_NCMBBUTTONUP	가운데 버튼을 떼를 때
WM_NCMBBUTTONDOWNDBLCLK	가운데 버튼을 더블 클릭할 때
WM_NCRBUTTONDOWN	오른쪽 버튼을 누를 때
WM_NCRBUTTONUP	오른쪽 버튼을 떼를 때
WM_NCRBUTTONDOWNDBLCLK	오른쪽 버튼을 더블 클릭할 때
WM_NCMOUSEMOVE	마우스를 움직일 때

Mouse Input

- 비 client 영역 mouse message handler

메시지	메시지 맵 매크로	메시지 핸들러
WM_NCLBUTTONDOWN	ON_WM_NCLBUTTONDOWN()	OnNcLButtonDown
WM_NCLBUTTONUP	ON_WM_NCLBUTTONUP()	OnNcLButtonUp
WM_NCLBUTTONDBLCLK	ON_WM_NCLBUTTONDBLCLK()	OnNcLButtonDbLCLK
WM_NCMBBUTTONDOWN	ON_WM_NCMBBUTTONDOWN()	OnNcMButtonDown
WM_NCMBUTTONUP	ON_WM_NCMBUTTONUP()	OnNcMButtonUp
WM_NCMBUTTONDBLCLK	ON_WM_NCMBUTTONDBLCLK()	OnNcMButtonDbLCLK
WM_NCRBUTTONDOWN	ON_WM_NCRBUTTONDOWN()	OnNcRButtonDown
WM_NCRBUTTONUP	ON_WM_NCRBUTTONUP()	OnNcRButtonUp
WM_NCRBUTTONDBLCLK	ON_WM_NCRBUTTONDBLCLK()	OnNcRButtonDbLCLK
WM_NCMOUSEMOVE	ON_WM_NCMOUSEMOVE()	OnNcMouseMove

Mouse Input

- 비 client 영역 mouse message handler 형태

```
afx_msg void OnNc##### (UINT nHitTest, CPoint point);
```

- nHitTest: Message가 생성될 때 mouse cursor 위치를 나타내는 상수
- point: Message가 생성될 때 mouse cursor 위치(screen 좌표)
 - Client 영역 좌표로 변환하려면 CWnd::ScreenToClient() 함수를 사용
- nHitTest 상수값

상수값	의미
HTCAPTION	타이틀바
HTCLIENT	클라이언트 영역
HTCLOSE	종료 버튼
HTHSCROLL	가로 스크롤 바
HTMENU	메뉴
HTMAXBUTTON 또는 HTZOOM	최대화 버튼
HTMINBUTTON 또는 HTREDUCE	최소화 버튼
HTSYSTEMMENU	시스템 메뉴
HTVSCROLL	세로 스크롤 바

Mouse Input

- 기타 mouse 관련 함수
 - Mouse cursor의 모양 지정:

```
HCURSOR SetCursor(HCURSOR hCursor);
```

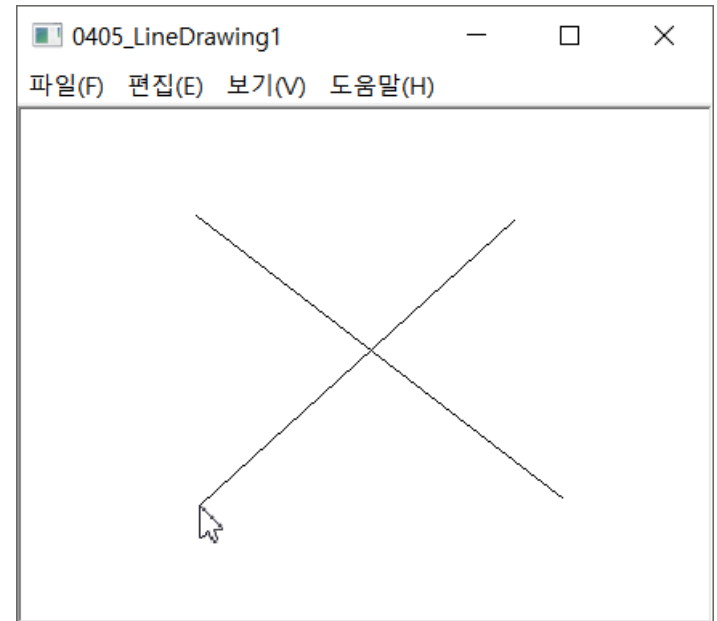
- Mouse cursor의 동작 범위 지정

```
Bool ClipCursor(const RECT *lpRect);
```

Mouse Input

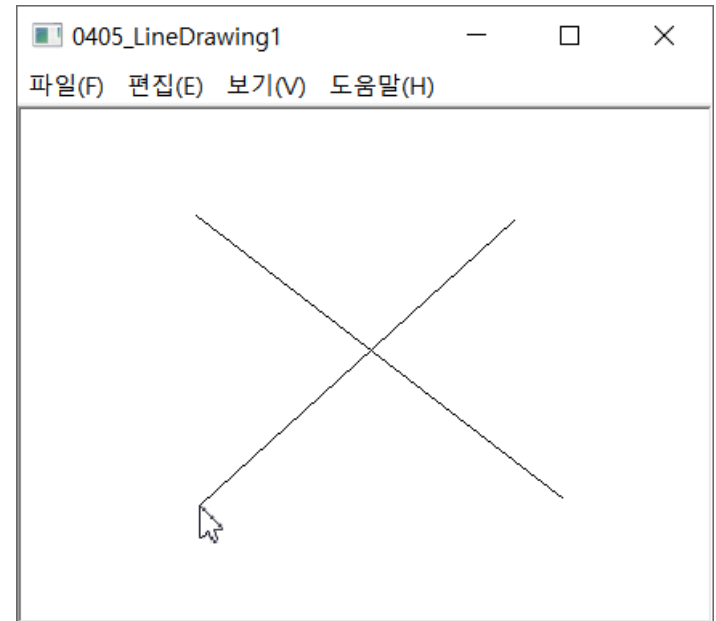
- 사용 예: Mouse를 이용하여 line 그리기 (1/2)
 1. Mouse button down이면, 시작점을 입력
 2. Mouse button up이면, 끝점을 입력하고 line 그리기

→ Line을 그리는 중간과정을 출력?



Mouse Input

- 사용 예: Mouse를 이용하여 line 그리기 (2/2)
 1. Mouse button down이면, 시작점을 입력
 2. **Mouse move**이면, 끝점을 입력하고 **line** 그리기
 3. Mouse button up이면, 끝점을 입력하고 line 그리기

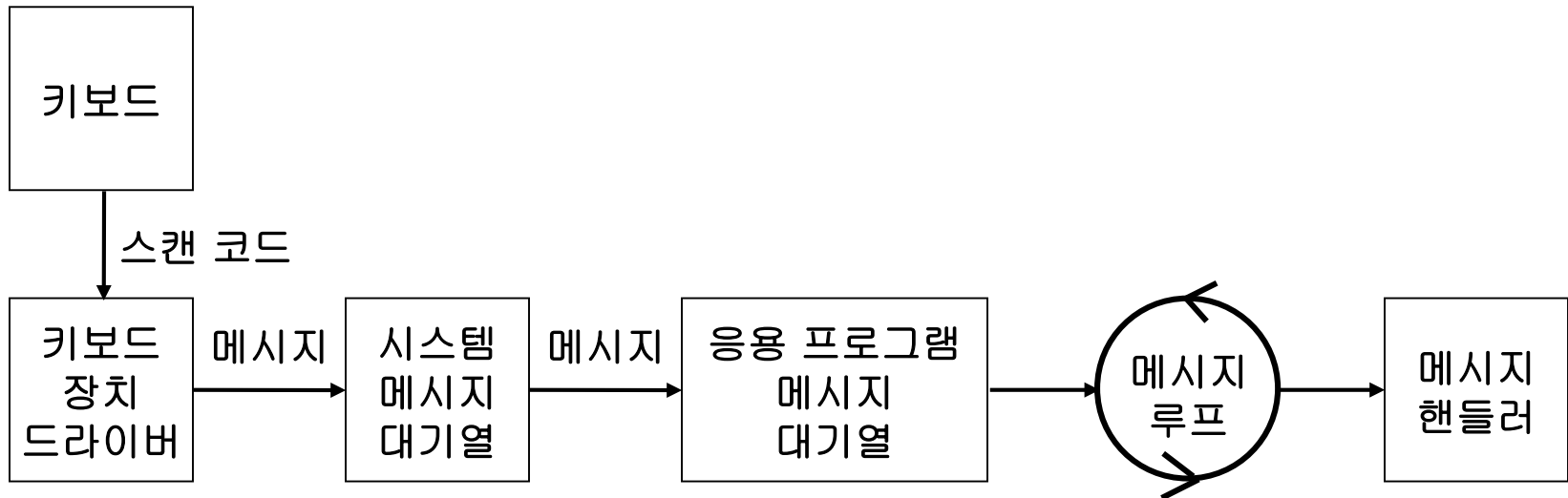


MFC와 Keyboard



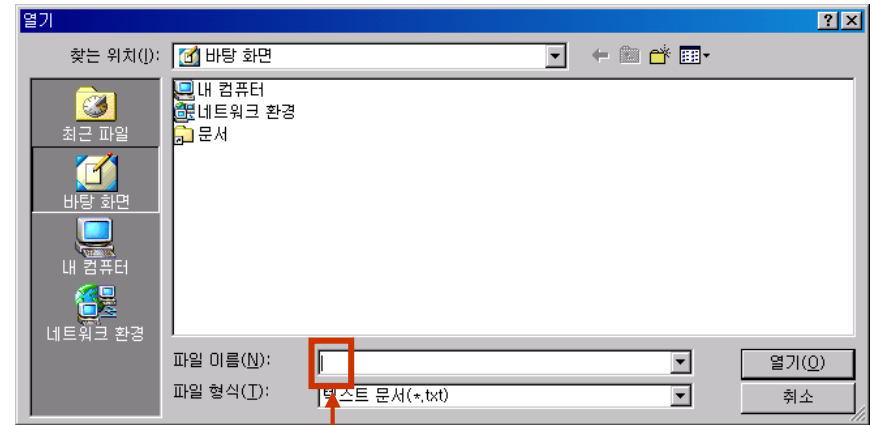
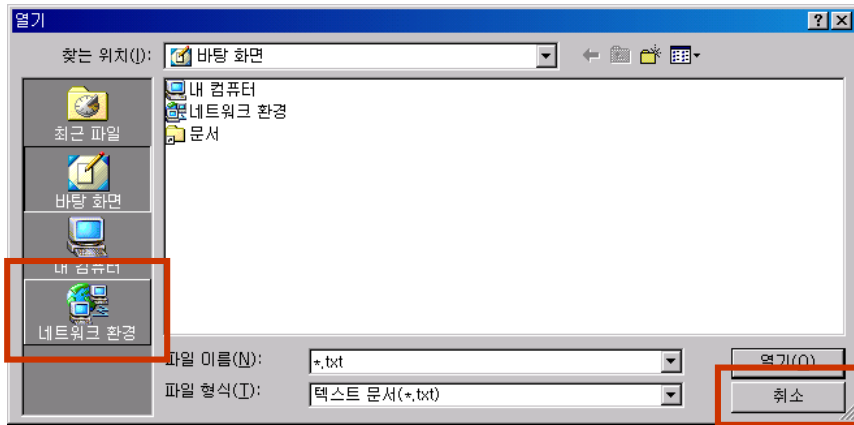
Keyboard Input

- Keyboard 처리
 - 윈도우 운영체제는 keyboard와 관련된 모든 변화를 message 형태로 program에게 전달



Keyboard Input

- Keyboard message 전달
 - Keyboard message는 keyboard focus를 가진 윈도우가 받음
- Keyboard focus
 - 활성 윈도우 또는 활성 윈도우의 자식 윈도우가 가지는 일종의 속성
 - Keyboard focus 유형

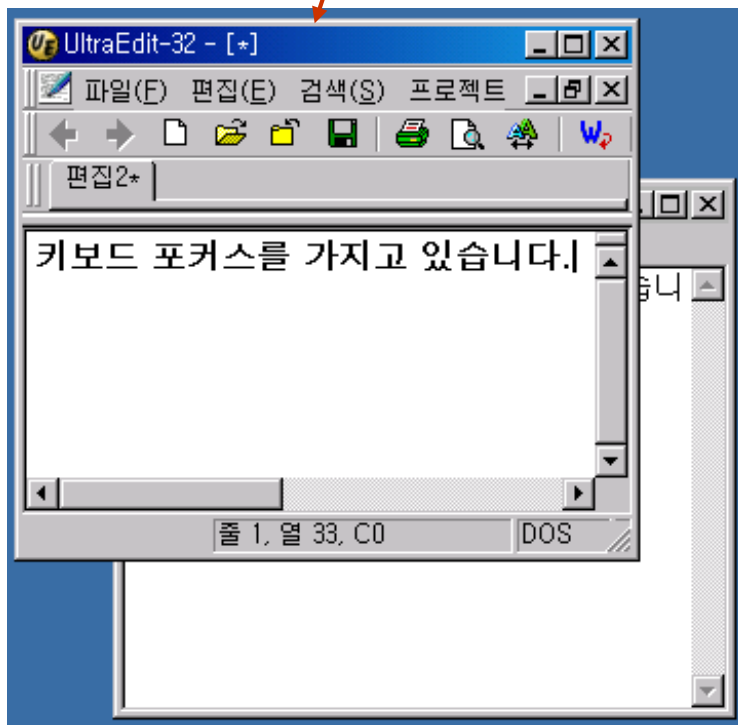


Caret(캐럿)

Keyboard Input

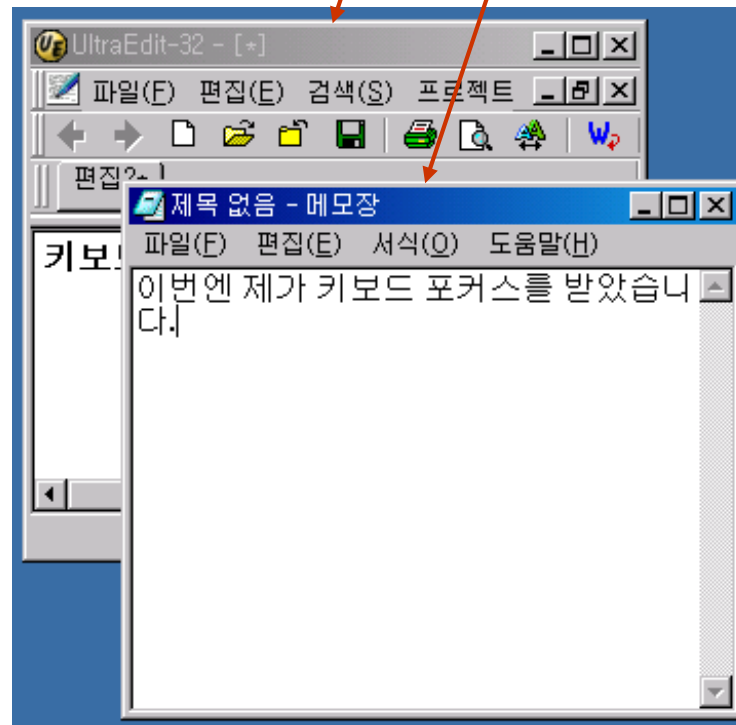
- Keyboard focus 변화

WM_SETFOCUS



WM_KILLFOCUS

WM_SETFOCUS



Keyboard Input

- Caret 관련 함수

함수 이름	역할
CreateCaret()	비트맵을 이용하여 캐럿을 생성한다.
CreateGrayCaret()	회색 사각형 모양의 캐럿을 생성한다.
CreateSolidCaret()	검정색 사각형 모양의 캐럿을 생성한다.
ShowCaret()	캐럿이 보이도록 한다.
HideCaret()	캐럿을 숨긴다.
GetCaretPos()	캐럿의 위치(클라이언트 좌표)를 얻는다.
SetCaretPos()	캐럿의 위치(클라이언트 좌표)를 변경한다.
::DestroyCaret()	캐럿을 파괴한다.
::GetCaretBlinkTime()	캐럿이 깜박이는 간격을 얻는다.
::SetCaretBlinkTime()	캐럿이 깜박이는 간격을 설정한다.

MFC에서 제공하지 않는 기능은 API 함수를 직접 호출

Keyboard Input

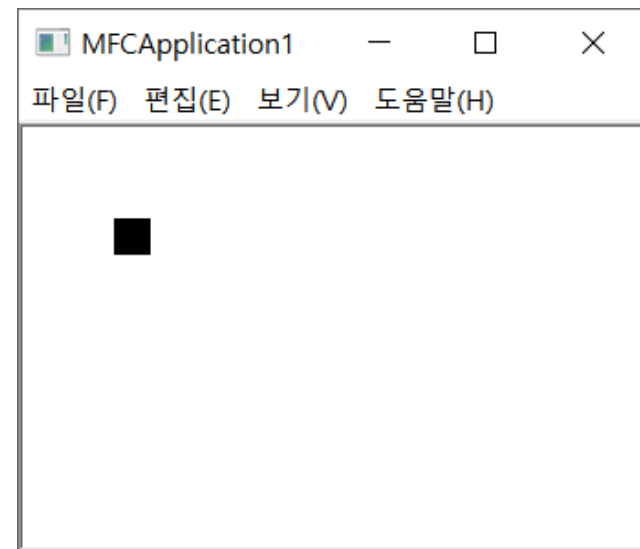
- 사용 예: Caret 관련 함수

```
void CChildView::OnSetFocus(CWnd* pOldWnd)
{
    CWnd::OnSetFocus(pOldWnd);

    CreateSolidCaret(20, 20);
    SetCaretPos(CPoint(50, 50));
    ShowCaret();
}

void CChildView::OnKillFocus(CWnd* pNewWnd)
{
    CWnd::OnKillFocus(pNewWnd);

    HideCaret();
    ::DestroyCaret();
}
```



Keyboard Input

- Keystroke(키 누름) message
 - Key를 누르거나 떼는 동작에 의해 발생하는 message
- Keystroke message 종류

메시지	의미
WM_KEYDOWN	F10, Alt 이외의 키를 누를 때
WM_KEYUP	F10, Alt 이외의 키를 떼를 때
WM_SYSKEYDOWN	F10, Alt, Alt + [키 조합]을 누를 때
WM_SYSKEYUP	F10, Alt, Alt + [키 조합]을 떼를 때

Keyboard Input

- Keystroke message handler 형태

```
afx_msg void On####(UINT nChar, UINT nRepCnt, UINT nFlags);
```

- nChar: Key에 할당된 virtual key code 값 (아래표 참조)
- nRepCnt: Key를 계속 누르고 있을 경우 1보다 큰 값을 가짐
- nFlags: Key와 관련된 다양한 정보를 담고 있음 (MSDN 참조)

가상 키 코드	해당 키	가상 키 코드	해당 키
VK_CANCEL	Ctrl-Break	VK_HOME	Home
VK_BACK	Backspace	VK_LEFT	←
VK_TAB	Tab	VK_UP	↑
VK_RETURN	Enter	VK_RIGHT	→
VK_SHIFT	Shift	VK_DOWN	↓
VK_CONTROL	Ctrl	VK_SNAPSHOT	Print Screen
VK_MENU	Alt	VK_INSERT	Insert
VK_PAUSE	Pause	VK_DELETE	Delete
VK_CAPITAL	Caps Lock	'0' - '9'	0 ~ 9
VK_ESCAPE	Esc	'A' - 'Z'	A ~ Z
VK_SPACE	Spacebar	VK_F1 ~ VK_F12	F1 ~ F12
VK_PRIOR	PgUp	VK_NUMLOCK	Num Lock
VK_NEXT	PgDn	VK_SCROLL	Scroll Lock
VK_END	End	VK_SCROLL	Scroll Lock

Keyboard Input

- Character(문자) message (1/2)
 - 키 누름 message 사이에 발생하는 WM_*CHAR message
 - 필요성: VK_R 키를 누른 경우

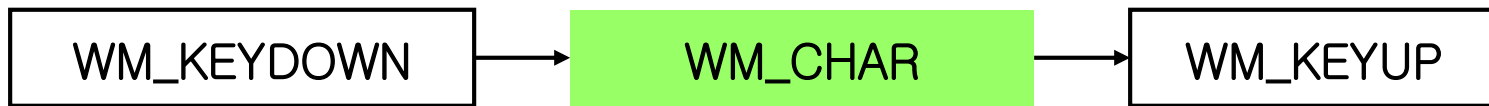
문자	가상 키 코드 조합
r	영문 입력 모드에서 VK_R 또는 Caps Lock + Shift + VK_R 키를 누른 경우
R	영문 입력 모드에서 Caps Lock + VK_R 또는 Shift + VK_R 키를 누른 경우
ㄱ	한글 입력 모드에서 VK_R 키를 누른 경우
ㄲ	한글 입력 모드에서 VK_R + Shift 키를 누른 경우

→ 키 누름 메시지 (WM_KEYDOWN) 외에 문자 메시지 (WM_CHAR)를 발생 시킴

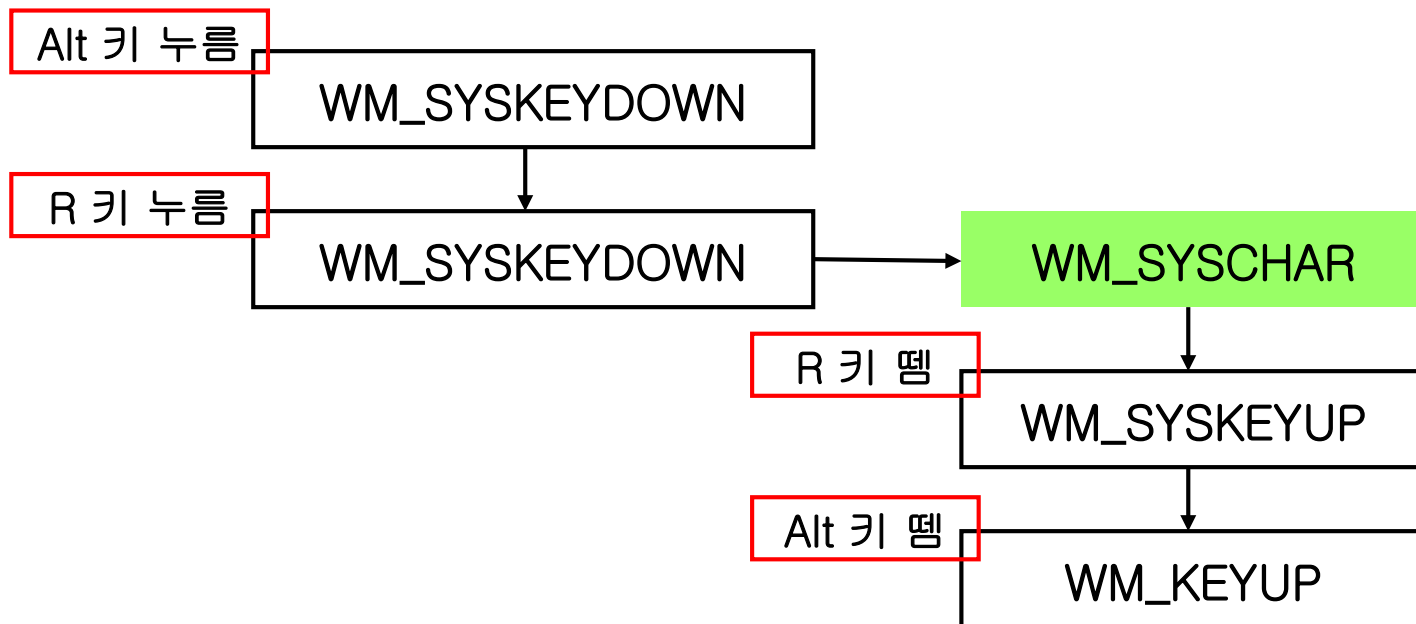
Keyboard Input

- Character(문자) message (2/2)

- VK_R 키를 누른 경우



- Alt + VK_R 키를 누른 경우



Keyboard Input

- Character(문자) message handler

```
afx_msg void OnChar(UINT nChar, UINT nRepCnt, UINT nFlags);  
afx_msg void OnSysChar(UINT nChar, UINT nRepCnt, UINT nFlags);
```

- nChar: Key에 해당하는 character code 값을 가짐
- nRepCnt: Key를 계속 누르고 있을 경우 1보다 큰 값을 가짐
- nFlags: Key와 관련된 다양한 정보를 담고 있음 (MSDN 참조)

*키 누름 message로는 대소문자(예: 'A' 또는 'a')를 구분할 수 없음. 따라서, 문자 message의 character code 값(예: nChar의 ASCII 코드 값)을 사용해야함

Keyboard Input

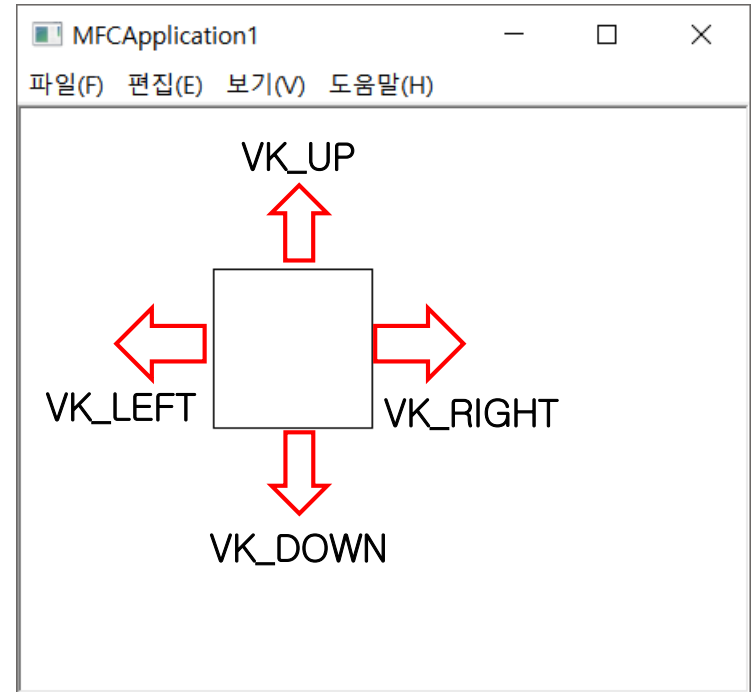
- 사용 예: 사각형의 위치를 방향키로 조작하기

```
// 1. 위치 저장 변수 선언
CPoint m_pt;

// 2. 위치에 사각형 그리기
dc.Rectangle(m_pt.x, ...);

// 3. WM_KEYDOWN handler 추가
void CChildView::OnKeyDown();

// 4. nChar 값 검사하여 위치변경
if (nChar == VK_RIGHT)
    m_pt.x += 10;
```



Q & A